EXPRESS MAIL NO.: EL 754 409 645 US

International Business Machines Corporation Docket No.: YOR920010074US1

Ohlandt, Greeley, Ruggiero & Perle, L.L.P. Docket No.: 909.0043 USU

Patent Application Papers of: David Bantz

Steven Mastrianni

METHOD AND APPARATUS FOR PERFORMING EMERGENCY SHUTDOWN OF A MALFUNCTIONING COMPUTER SYSTEM SAVING ALL OPEN FILES, DATA, AND WORK IN PROGRESS TO A REMOTE DATA STORAGE BUSINESS ENTITY

CROSS-REFERENCE TO A RELATED PATENT APPLICATION:

This patent application is related to U.S. Patent Application S.N. 09/_____, filed on even date herewith, entitled "METHOD AND APPARATUS FOR PERFORMING EMERGENCY SHUTDOWN OF A MALFUNCTIONING COMPUTER SYSTEM SAVING ALL OPEN FILES, DATA, AND WORK IN PROGRESS", by Steven Mastrianni and David Bantz (Attorney Docket No.: YOR920010070US1).

FIELD OF THE INVENTION:

This invention relates generally to digital data processing systems and methods and, more particularly, to methods and apparatus for ameliorating the adverse effects of data processing system malfunctions.

BACKGROUND OF THE INVENTION:

As computer operating systems have grown more complex and feature-rich, they have also grown significantly larger. In order to more effectively manage these large software systems the functionality of the operating system has been partitioned into separate modules or components. The underlying rationale in partitioning the operating system is that dividing up a complex software system into smaller pieces makes the operating system easier to develop and maintain. While this may be true to a first order, the fact is that personal computer (PC)-based operating systems such as, but not limited to, Microsoft Windows 95™, Microsoft Windows 98™, and Microsoft Windows 2000™ all experience "crashes" or "hangs",

requiring the user to reboot the PC, or in some cases to cycle the power off and then on in order to restore the PC to an operational status.

It should be noted that not all of these problems are the fault of the operating system, and may instead actually be the fault of a running application. For example, the applications programmer may have forgotten to test a program path, or a program path was never traversed during testing so that some problem did not appear. In some cases, faults in the application may simply be the result of poor programming, or the failure to check the success or failure of a particular function call. Whatever the reason, the result is that the user's PC can become unresponsive or non-responsive and effectively unusable until the system is rebooted or powered off and then back on. As many user's have experienced, such problems typically occur at the most inopportune times, such as when a document or a spreadsheet must be quickly completed in order to make a deadline. Furthermore, the occurrence of these types of system failure events is unpredictable, and may occur at any time during use of the operating system or application.

One common failure mode is that a program is "leaking" memory, i.e., gradually using up small amounts of memory without releasing the used portions. This results in physical RAM memory capacity being slowly consumed until there is no more available. This type of failure mode typically causes the operating system to stop responding to user input such as keyboard strokes or mouse clicks. Because this type of failure occurs slowly over time, the user may have been working on a document or presentation for quite a while, and may have made many changes or additions to the work in progress. When the system stops responding, the user will be unable to save the work in progress, resulting in a high probability that the work in progress will be lost. In order to restore the system to a functional state the user is typically required to reboot or power cycle the system, resulting in the loss of all or at least some edits to the work in progress being lost.

Another less common problem occurs when a program consumes most or all of the computer's processing capability, thereby starving other programs of computer time for proper execution. In this situation, the system may not respond to the user's mouse movements and certain keyboard commands, including the command to restart the system,

requiring the user power cycle the system in order to restore the system to a functional state.

The foregoing problems are not limited to only user applications, but apply as well to operating systems themselves. Existing operating systems require an orderly closing of applications. As each application is closed the user is presented with a dialog box asking whether changed data is to be preserved. The operating system itself requires orderly shutdown to flush its file caches, save user settings, close network connections and the like. If operating system processes have to be terminated abnormally, dialogs are presented to the user in each case. Shutdown from a moderately complex operating configuration can take upwards of a minute and require multiple user interactions. The inventors are not aware of any case wherein the operating system or an application take account of the need to move data out of the computer system to a safe place.

As can be appreciated, a need exists to lessen and ideally avoid the negative impact of system failures, malfunctions and unforeseen events on the user's work environment.

SUMMARY OF THE INVENTION

The foregoing and other problems are overcome by methods and apparatus in accordance with embodiments of this invention. The teachings of this invention provide methods and apparatus to enable an orderly shutdown of a malfunctioning computer system, while saving user-desired work in progress. In the presently preferred embodiment of this invention the user-desired work in progress, including open files, is transferred over a data communications network, such as the Internet, from the computer system to a remote data storage business entity. Subsequently, the stored work in progress is transferred back to the computer system for restoring the user as closely as possible to the point where the user's work was interrupted.

In one aspect these teachings provide a software program and software device driver mechanism that is installed on a computer system to be protected. In accordance with these teachings the software program "hooks" or connects or links to the operating mechanism that is the single point of entry for a file open request. When a program makes a request to open a

file, the request is detected, and the name of the requesting program is identified and saved along with other information, including the name of any file or files that the program is requesting to open. The data is saved as a protected database in a protected area of system memory, preferably but not necessarily a local disk drive.

The software program also "hooks" or connects or links to the single entry point for a file close function. Each time a program closes a fil the close request is intercepted. The name of program closing the file is detected, along with other information, including the name of the file or files to be closed. The files are closed normally, and the protected database updated with the new information.

Each time a file is opened or closed, the protected program/file database on the disk is updated, as well as a copy stored in computer memory (RAM). When the system stops responding to user input, the offending program may still have one or more files open, and may be unable to respond to any user request to properly close the program and the open file(s). When this occurs, the user invokes the emergency shutdown procedure in accordance with these teachings by using a predetermined "button". The button may be located on the keyboard, or it may be mounted internally or externally from the computer system. The button may be implemented using a conventional type of pushbutton or other type of switch, or it may be implemented using some predetermined keyboard key sequence where the user depresses one or more keys, either sequentially or together. The special key combination is detected, and is used to invoke the operation of the orderly shutdown procedure in accordance with these teachings.

In operation, the orderly shutdown procedure enumerates the list of currently running programs and then by using a program state table determines which program or programs are no longer responding to user input. In accordance with a further aspect of these teachings, the resulting list of non-responding programs is compared to a list of non-responsive programs, i.e., a list of programs that are normally in a non-responsive state, to determine which user program or programs are not responding. The list of non-responsive programs preferably includes those programs that are part of the operating system and that normally run in a suspended or "not responding" mode.

After the identities of the user non-responding programs are determined, the orderly shutdown procedure attempts to read from the protected database to determine if any of the non-responding user programs have any files open. If the disk is not accessible, the information is read from the copy stored in the memory-resident database. The orderly shutdown procedure matches each non-responding program to any open file or files that it may have, and transfers a copy of the identified file or files to a remote data storage business entity along with, by example, a date and time stamp, as well as the name of the non-responding program. When the open file transfer procedure terminates, the orderly shutdown procedure may display a message box indicating that any open files have been saved, and that it is safe for the user to shutdown the system. It is also within the scope of these teachings to provide for the local storage of the open file(s) at the computer system, in a manner described in the above referenced related U.S. Patent Application.

After the computer system is rebooted, the user requests that the transferred file or files be retrieved from the remote data storage business entity, and may then resume working at the point (ideally) where the work was stopped.

In a further aspect the teachings of this invention apply as well to a data storage business entity or service that provides for the storage of files and related information for computer system users that have experienced a computer system malfunction, or that for some other reason desire to store their work in progress and related information.

This further aspect of these teachings provides a business model based on providing a service that aids the user in performing an emergency or "panic" shutdown and, equally importantly, in restarting after a panic shutdown. A value of the service provided is the preservation of potentially key user data across a panic shutdown and restart. The foregoing and other situations (e.g., an occurrence of external events, potentially catastrophic internal events, etc.) will often imply that the data that must be preserved must be quickly moved from the user's computing system to another location. It is this movement, and the reverse migration, that the business service provides.

In accordance with these teachings a method is disclosed for operating a digital data processing system, as is a digital data processing system that operates in accordance with the method. The method includes steps of (A) detecting an activation of a user input that indicates that the system or a program executed by the system has become non-responsive to the user; (B) in response to detecting the activation of the user input, transferring to a data storage business entity, through a data communications network, selected user information; and (C) operating the data storage business entity so as to store the transferred user information, and to subsequently retrieve and transfer back to the digital data processing system at least some of the stored user information.

Also disclosed is a data storage business entity having a data communications network interface for coupling the data storage business entity to client data processing systems and a controller that is responsive to an arrival of information from a client data processing system for storing the information and for subsequently retrieving and transferring at least some of the retrieved information back to the client data processing system. The controller further operates to charge the client data processing system for at least one of storing the user information, retrieving the stored information, or for being available to store and retrieve the user information.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features and advantages of the present invention will be apparent to one skilled in the art, in view of the following detailed description taken in combination with the attached drawings, in which:

Fig. 1 illustrates a hardware block diagram depicting the orderly shutdown procedure in accordance with these teachings;

Fig. 2 is a logic flow diagram depicting the operation of the orderly shutdown procedure; and

Fig. 3 is a block diagram that illustrates in further detail the network-connected data storage business entity shown in Fig. 1.

DETAILED DESCRIPTION OF THE INVENTION

Referring to Fig. 1 a computer system 100 is illustrated that is suitable for practicing this invention. The computer system 100 may be a PC, or it may be a workstation, or a minicomputer or a main frame computer. The computer system 100 may also be integrated within some other type of device, such as a communications device, or a personal digital assistant (PDA), or an input device that digitizes a user's handwriting. As should be appreciated, the teachings of this invention may be used in any computing environment wherein a user is enabled to input information to the computer system 100, and where it is desirable to maintain and recover the entered information in the event of a computer system malfunction.

The exemplary computer system 100 may include at least one central processing unit (CPU) 110, a read/write memory 120, such as RAM or flash memory, a mass storage device such as a fixed or a removable disk 130 and/or a tape, a user interface 140 that includes a display 142, a keyboard 144 (or some other data entry device), and a pointing device 148, such as a mouse. A network interface 150 is provided for coupling the computer system 100 to an external network 160, such as an intranet, an extranet, the Internet, or any type of data communications network.

In accordance with the teachings of this invention, connected to the network 160 is a data storage business entity 165, which is assumed, as shown in Fig. 3, to contain or control data mass storage devices 200, such as disk drives and/or tape drives, and a data processor 210 that is responsive to the computer system 100 for storing and retrieving files and related information. These aspects of the invention are described in further detail below.

Still referring to Fig. 1, the memory 120 is assumed to store all or part of the operating system (OS) 122, as well as relevant device drivers and tables 124, as well as any desired applications 126. The applications could include, by example, a word processing applications, a spreadsheet application, a visual slide preparation application, a financial application, and/or any type of desired application. When not in use the program code that

implements these applications may be stored on the fixed disk 130.

In accordance with an aspect of these teachings the user interface 140 includes a button 146 that is activated by the user when it is desired to perform an orderly shutdown of one or more running applications 126, and to save all work that may be in progress (e.g., a document or a spreadsheet that is being composed or revised). The button 146 is shown for convenience as being located at the keyboard 144, although it may be located anywhere that is convenient for the user. The button 146 may be physically implemented using a conventional type of pushbutton or other type of switch, or it may be logically implemented using some predetermined keyboard key sequence where the user depresses one or more keys, either sequentially or together. Further by example, if the user interface 140 includes a voice recognition capability, the button 146 may be implemented by recognizing some predetermined user utterance. Also by example, if the user interface 140 includes a handwriting recognition capability, then the button 146 may be implemented, for example, by recognizing some predetermined sequence of strokes, or by recognizing stoke input made at some predetermined position on a handwriting tablet.

It is assumed that the button 146 is activated by the user when the computer system 100 ceases to respond to user inputs, e.g., when the system "freezes", or when the system provides some error message that it is unable to continue to operate. That is, it is assumed that the at least one program or application with which the user is interacting has ceased to accept the user's input, or to otherwise behave in a fashion that is inconsistent with the user's expectations. It is also within the scope of these teachings to activate the button 146 should the user perceive some threat (e.g., fire, earthquake), or should the user feel ill and unable to continue working, or should the user notice some computer system anomaly, such as a virus erasing files, or an unusual sound, or smoke. The use of the teachings of this invention are particularly useful in such situations, as it may be especially desirable to transfer the user's work in progress to a remote location, i.e., away from the potential hazard that may exist at the computer system 100.

However the button 146 may be implemented, when the user activates the button 146 a signal is sent via a suitable digital or analog interface to the CPU 110. For example, the button

interface may convert the button signal into a system interrupt (INT). The interrupt, which is preferably installed at a high priority (and which may be non-maskable), invokes a software module (SM) 128A that cooperates with the data storage business entity 165 to begin operating to save the relevant data files.

Referring now also to Fig. 2, at Step A the interrupt is received by the software module 128A and is routed to the device driver 124. At Step B the device driver 124 notifies the application 126 which enumerates the list of programs and their open files at Step C. At Step D the software module 128A transfers the information to the data storage business entity 165 over the network 160, where the transferred information is saved, and possibly also saved in a local memory, such as in a (protected) database 132 on the fixed disk 130 and/or in the memory 120. At Step E the software module 128A notifies the user that it is safe to shut down or power off the computer system 100, such as by notifying the user that the information, including open files, has been transferred to the data storage business entity 165. In a Step F the saved information, such as the previously open files, are retrieved from the data storage business entity and restored to the system memory of the computer system 100.

Describing Fig. 2 now in further detail, and prior to the execution of the Steps A-F, i.e., during normal operation of the computer system 100, an element of the software module 128A "hooks" or connects or links to the operating mechanism that is the single point of entry for a file open request. When a program or application 126 makes a request to open a file, the request is detected and the name of the requesting program or application 126 is identified and saved along with other information, including the name of any file or files that the program is requesting to open. This information is saved as in the protected database 132. Optionally the copy of the protected database 128B is stored in the memory 120.

The software module 126A also "hooks" or connects or links to the single entry point for the file close function. Each time a program or application 126 closes a file the close request is intercepted. The name of program or application 126 closing the file is detected, along with other information, including the name of the file or files to be closed. The files are closed normally, and the protected database(s) 132, 128B are updated with the new information. In this manner the protected database 132 reflects, at any point in time, only those files that are

currently open, as well as the application 126 with which the open files are associated.

This being the case, at Step C of Fig. 2 the orderly shutdown procedure executed by the software module 128A enumerates the list of currently running programs and then, by using a program state table maintained by the operating system 122 that forms a part of the drivers and tables 124, determines which program or programs are no longer responding (NLR) to user input. The resulting list of NLR programs is compared to a list of normally unresponsive programs (UPs) 128C, i.e., a list of programs that are normally in an unresponsive state. If an NLR program is found in the UP list no further action need be taken, since the program is normally in an unresponsive state, and its lack of responsiveness is thus normal. Examples of programs that may be found on the NLR list, but would not normally be found on the UP list, are Lotus Notes™, Microsoft Word™, Adobe Photoshop™, Microsoft Publisher™ and Microsoft Powerpoint™. Examples of programs that would normally be found on the UP list include Norton Antivirus Services™, WINLOGON, and various communication services.

The list of UP 128C preferably includes those programs that are part of the operating system 122, and possibly those associated with installed applications that normally run in a suspended or "not responding" mode of operation. The list of unresponsive programs 128C may be established at the time the operating system is installed, and can be updated as required when changes are made to the operating system 122, and/or when new applications are installed.

After the identities of the user non-responding programs are determined, the orderly shutdown procedure executed by the software module 128A attempts to read from the protected database 132 to determine if any of the non-responding user programs have any files open. If the fixed disk 130 is not accessible, the information is read from the copy 128B stored in the memory-resident database. The orderly shutdown procedure executed by the software module 128A then matches each non-responding program to any open file or files that it may have. During Step D the software module 128A operates to transfer a selected file or files to the data storage business entity 165, while preferably identifying these files with, by example, a date and time stamp, the name of the non-responding program and any other pertinent information. Depending on the application, other information may be available and

may be stored, such as the last n keystrokes and/or blocks of copied text of a word processing application program. Some or all of this information may also be stored locally, such as on the fixed disk 130. When the open file transfer procedure terminates, the orderly shutdown procedure executed by the software module 128A may display a message box at Step E for indicating that any open files have been saved, and that it is safe for the user to shutdown the system.

After the computer system 100 is rebooted, at Step F the user is enabled to request that the saved files and other information be retrieved from the data storage business entity 165, at which time the requested files and other information are transferred over the network 160, through the network interface 150, and are restored to the system memory by the operating system122. The user may then resume working at the point where work stopped, which is the desired result.

Depending upon the characteristics of the operating system 122 (e.g., Windows™, Linux™, Unix™) various different types of techniques may be employed for obtaining information regarding the applications or processes and their files. As an example, the Windows NT™ and Windows 2000™ approach to creating a list of processes and modules can be found from publically-available help posted on the Microsoft website. Briefly, this approach uses functions from a PSAPI.DLL file. The PSAPI.DLL file is distributed with the Platform SDK, available at: http://www.microsoft.com/msdn/sdk. PSAPI.DLL is the library used to gather process information on Windows NT™ and resides in the \SYSTEM32 directory.

In a manner similar to ToolHelp32 functions, the PSAPI.DLL also contains a variety of useful functions. Those functions that are relevant to enumerating processes and modules include:

EnumProcesses()
EnumProcessModules()
GetModuleFileNameExA()

Briefly, a call is first made to EnumProcesses() to fill an array of process Ids. A ModList

sample code example also includes a method of calculating the number of processes returned.

Next, OpenProcess() is called for each of the process IDs to retrieve a handle to the process, and if the handle is valid, then a call is made to EnumProcessModules() to enumerate the modules of the process. EnumProcessModules() fills an array, passed as a parameter, with the module handles associated with the process.

GetModuleFileNameExA() is used to retrieve the name of the module using the process handle and module handle as parameters. The module name would be the path and file name of the dll, ocx, etc., that the process has loaded.

It is noted that the name of a process may also be displayed in the list of modules for that process. If this is not desirable, then one may simply compare the module name to the process name before adding it to the list.

The on-line help mentioned above contains a Visual Basic™ example that is useful for illustrating how one may list the processes that are currently running on the computer system 100. The approach is different for Windows 95/98 and Windows NT/2000, and both solutions are described in the on-line help.

An important aspect of these teachings is a method and software architecture for emergency shutdown of active applications. Although this is valuable as a self-contained mechanism for reacting to a panic situation, by complementing the mechanism with the service provided by a remote service provider, embodied in the data storage business entity 165, additional user benefit can be obtained. Specifically, the movement of user data away from the computer system 100 is a significant benefit in the case that the system 100 is, itself, physically threatened.

The business model by which the data storage business entity 165 operates provides a service, for a fee, that functions as an emergency remote backup, capturing key end-user data. This data may include, but is not limited to, open work files currently associated with running applications, open files associated with the operating system, and user settings and

personalizations such as bookmarks. The service is valuable for the copying of this data remotely in response to a physical threat; it is equally valuable for the restoration of this data from a remote site, selectively, so that normal operation of the computer system 100 can be resumed after an emergency or panic shutdown.

The general disposition of the components of an implementation of the data storage business entity 165 is shown in Fig. 3. In this figure the user is shown at the computer system 100 with the dedicated button 146, also now referred to as a Panic Button. The user and computer system 100 are assumed to be one of many clients of the data storage business entity 165. The user's computer system 100 communicates via network 160 and a network interface (NI) 202 with a panic button service 205 of the data storage business entity 165 in order to save and restore the user's and other data. The user may also contact an operator 220 to assist in the restoration of data after an emergency shutdown. A fulfillment and billing component 215 determines whether the user and/or the computer system 100 are subscribed to the service, and based on a charging algorithm, issues debits against the user's account for use of the panic button service 205. The fulfillment and billing component 215 may also run on the data processor 210, or it may run on another (local or remote) data processor.

The panic button service 205 may be part of a comprehensive "safety net" offering that includes backup, so that the data saved on emergency shutdown is just that data that has changed since the last backup. The offering can also provide readiness checks, user training, and policy-driven configuration.

The panic button service 205 may charge the user in several ways. For example, the user can be charged a fixed monthly fee for providing (and testing) the infrastructure necessary to provide the panic button service 205. The user may also be charged a per-incident amount for actually performing the data preservation service, as well as a per-incident charge for restoring data after an event. The charges may be related in some manner to the amount of data preserved/restored. There may also be charges for operator 220 intervention to assist in the process of restoring data, and for other actions related to recovery from an emergency shutdown.

It should be noted that the network 160 connecting the computer system 100 and the data storage business entity 165 can be a wired network, including an optical fiber network, telephone lines, coaxial cable and the like, or it may be a wireless network, such as one composed of one or more RF links. This latter embodiment is particularly useful when the computer system 100 is a portable, or is contained within some device having a wireless network interface.

The invention was described above in the context of a system and method for performing an orderly shutdown of a malfunctioning computer system, and the remote storage of user data. It should be appreciated that the teachings of this invention also relate to a computer readable medium that stores program instructions for directing the operation of CPU 110 in executing the orderly shutdown of the malfunctioning computer system. The computer readable medium can include the fixed disk 130, the memory 120, a removable disk or tape, a read only memory device, or any suitable computer readable medium. Portions of the computer readable medium are also associated with the data storage business entity 165 for directing same to operate as described above.

It should be noted that for the system to be considered non-responsive, it is not necessary for the system to have ceased to function. Instead, the functioning of the system may have degraded to the point where it becomes objectionable, or simply noticeable, to the user.

Thus, while the invention has been particularly shown and described with respect to preferred embodiments thereof, it will be understood by those skilled in the art that changes in form and details may be made therein without departing from the scope and spirit of the invention.